

A Science-Based Executive for Autonomous Planetary Vehicles

Stephen F. Peters

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109, USA
stevep@jpl.nasa.gov

Keywords robotic vehicle, rover, autonomy, intelligent execution, planetary exploration.

Abstract

The timing and success of execution of autonomous operations in natural environments is uncertain. Constraints such as round trip light time and deep space antenna scheduling can limit communications with a vehicle on a solar system body to one uplink per day. If requests for scientific observations, rather than specific plans, are uplinked to an autonomous execution system on the vehicle, it would be able to adjust its execution based upon actual performance, increasing science return for the day. Opportunistic science observations would also become possible. Such a science-based executive control system has been developed and demonstrated for the Rocky7 research rover.

1 Introduction

The performance, and indeed ultimate success, of operations executed by machines in contact with natural environments is uncertain. When the machine is a vehicle on a distant planet, light time alone can impose a feedback loop through an operator on Earth of tens of minutes or even hours. With other Earth-bound considerations such as deep space antenna scheduling, a reasonable expectation for planetary surface operations is one uplink per day.

During its day, the vehicle travels to various sites, positions its instruments, and takes measurements as requested by scientists on Earth. To maximize the effective use of the spacecraft's operational time, it is desirable for the spacecraft to have the ability to adjust its operations based upon actual performance that day.

Traditionally, requests for scientific operations are converted on the ground into a plan of timed actions which are then uplinked to the spacecraft. If the requests themselves are uplinked, the spacecraft can have the knowledge required to optimize operations based upon execution feedback. A system which supports this science-based uplink could also support opportunistic observation requests, triggered

by conditions sensed locally in the remote planetary environment.

With these goals in mind, a science-based executive control system has been developed and demonstrated with the Rocky7 research rover.

2 Knowledge Representation

To be able to adjust science operations in response to performance experienced on the planetary surface, the execution system must have knowledge of the science observations desired, the state of the vehicle, and the progress of attempted actions.

2.1 Science Requests

Three types of requests have been defined: nominal science observations which may be scheduled and executed, opportunistic science requests with associated trigger conditions, and safing actions with associated trigger conditions.

Nominal science requests contain the following information:

- the location in the local planetary surface frame where observation is to be made, specified as (x,y) where y points north and the distances are in meters;
- a relative ordering constraint, specified as *first*, *last*, or *anytime*, indicating when this request may be executed within the day's activity;
- relative priority, expressed as an integer;
- the amount of time the operation can nominally be expected to take, excluding time for traverse to location;
- the parameterized function call to be invoked to initiate the action or observation to be performed; and
- the current execution status of the request, specified as *request*, *active*, or *complete*.

Both opportunistic science requests and safing actions contain the following information:

- the parameterized function call to be invoked to initiate the action or observation to be performed and
- the function call to be invoked to monitor for trigger conditions and, when detected, issue the corresponding trigger event.

2.2 Rover State

Detailed information about rover state is maintained by the intelligent behaviors in Rocky7. The executive control system needs to reference a subset of this information:

most recently reported rover mobility state

- the current location in the local planetary frame, expressed as (x,y) where y points north and the distances are in meters;
- the current heading in the local planetary frame, expressed in radians west of north;

most recently reported mast and instrument state

- whether or not the mast is stowed;
- the current *pan* and *tilt* angles of the mast camera pair, expressed in radians;
- the current mast camera pair filter number; and

most recently reported mast and instrument state

- whether or not there is a sample in the scoop.

2.3 Progress of Execution

To manage the planning and execution of the day's science activity, the executive control system needs to maintain and monitor information about the progress of execution and reference additional information needed for planning:

states monitored by trigger conditions

- voltage on the power bus;
- voltages output by the sun sensor;

states related to the executing behavior

- whether or not a behavior is active;
- whether or not the active behavior can be interrupted;
- the last command issued to invoke a behavior;
- the status reported by the most recently completed or interrupted behavior;

parameters used for planning

- the average effective driving speed when driving while avoiding obstacles, expressed in meters per second; and
- a deadline for completion of nominal operations.

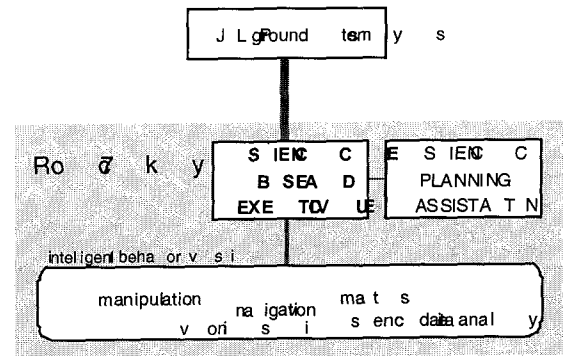


Figure 1: Rocky7 autonomy architecture.

3 An Autonomy Architecture

The autonomy architecture for Rocky7 [1] with the science-based executive is shown in Figure 1.

At the foundation of the Rocky7 autonomy architecture is a broad suite of robust intelligent behaviors implemented as finite state machines in ControlShell [2]. These range in complexity from simple actions such as driving wheels and taking images, to detection of navigation obstacles in stereo images, to driving to a specified location in the local planetary frame while avoiding obstacles. These behaviors provide feedback of success or failure in addition to updates to rover state.

A day's worth of scientific observation activity, based upon uplinked science requests, is composed from these behaviors by, and execution managed by, the science-based executive, assisted by a simple planner.

3.1 The Executive

The science-based executive is implemented using the Executive Support Language (ESL) [4], which was the basis for the executive in the Remote Agent Experiment [3] of the Deep Space One mission.

ESL provides a rich set of constructs for coordination and control of multiple concurrent software tasks. Tasks to achieve desired scientific operations are simply expressed, with necessary conditions monitored in independent concurrent tasks. Recovery actions to be performed in response to execution failure or external interrupt are also independent tasks.

The executive has a built-in function for each science request supported. At the heart of the executive is a simple function which monitors execution and reacts to trigger conditions, calling the planning assistant to replan from the set of uplinked requests as needed.

the executive algorithm

- Monitor for conditions indicating a safing action must be performed; when detected, interrupt active science operations and invoke the safing action.
- Meanwhile, perform science operations by repeating the following until no more outstanding science requests remain:
 - Make a nominal science operations plan from the set of outstanding science requests.
 - Monitor for conditions indicating an opportunity for making an opportunistic observation; when detected, interrupt the active nominal science operations plan and perform the opportunistic observation.
 - Meanwhile, perform the most recently made nominal science operations plan.

In this algorithm, if a nominal science operation fails, the entire nominal science operations plan fails, completing the current iteration of the repeat and proceeding to the next, resulting in a replan. If the nominal science operations plan completes with time to spare and science requests outstanding, a new plan is made, opportunistically making use of the unanticipated available time.

3.2 Interface with Behaviors

The intelligent behaviors are invoked by, and feedback is returned to, the executive via Network Data Delivery Service (NDDS) [5] messages. Queries to the behaviors are made as necessary to ensure consistency between the rover state used by the executive and the rover state maintained within the intelligent behaviors.

Science requests contain calls to functions defined within the executive rather than commanding the intelligent behaviors directly. These functions support a variety of initial states, issuing commands to behaviors at execution time as needed. These functions are easily invoked by the executive, and plans using them are easily constructed by the planner.

The commands to the intelligent behaviors and the executive functions which issue them are listed below.

commands to the intelligent behaviors

- calibrate the steering wheels
- calibrate the arm
- query behavior internal state
- go to (x,y) in the local planetary frame
- take a spectrometer reading
- acquire a sample
- select filter wheel
- stow the mast
- take an image
- point the mast camera pair

- interrupt driving

functions supporting science requests

- initialize for operations
- take a spectrometer reading
- acquire a sample
- take a panorama
- observe clouds
- safe the mast

Parameters to the function calls are included within the science requests and converted and passed as arguments to commands to intelligent behaviors.

3.3 A Simple Planner

In lieu of a full-featured planning assistant, a simple placeholder planner was implemented to select and order the set of science requests to form a nominal science operations plan.

Explicit ordering constraints, contained within the science requests themselves, are observed.

Implicit ordering constraints, derived from presence of sample in the scoop and dual use of the arm for taking spectrometer measurements, are also observed.

Since the duration of a traverse depends upon the start and end points of the traverse, which in turn depend upon the ordering of operations chosen by the planner, this duration is computed during the planning process, based upon a nominal speed for traversing while avoiding obstacles. Durations for the science operations themselves are included within the uplinked science requests.

Finally, all operations are planned to complete before the uplinked deadline.

Alternative plans are considered, discarding lowest priority requests first, until a plan meeting the temporal constraints is found.

The resulting executable plan is a simply ordered list of science requests. Function calls within the requests are invoked directly by the executive.

4 Test Operations

Many tests with Rocky7 have been performed, and a demonstration was videotaped in JPL's Mars Yard.

The science-based executive was given a deadline and a set of requests:

standard science requests

- acquire a sample
- make a spectrometer measurement
- take a panorama
- rover startup initialization

opportunistic science request

- observe clouds -- triggered by reduced sun sensor voltage

safing operation

- stow mast -- triggered by low power voltage

The executive called the planning assistant to make a plan which could be executed in the available time, estimating the amount of time required for scheduled observations and traverses between, and obeying other constraints.

initial plan

- rover startup initialization
- make a spectrometer measurement
- acquire a sample
- take a panorama

Initialization was executed and then Rocky7 started a traverse to the site for the spectrometer measurement. During this traverse, a reduced-sunlight event was issued, triggering the opportunistic observation. The traverse was interrupted, the mast deployed and pointed in the direction of the sun, an image was taken, and the mast was stowed. Ready to resume normal operations, the planning assistant was again called.

replan

- acquire a sample
- take panorama

There was no longer enough time to perform all the requested observations, so the planner skipped the lower-priority spectrometer measurement. Execution resumed with this new plan. During the panorama, a low-power event was issued, triggering the safing operation. The panorama was interrupted, the mast stowed, and Rocky7 was left in an idle state.

The resultant execution is depicted in Figure 2. The dotted arrows represent the initial plan and the solid arrows represent the actual operations performed.

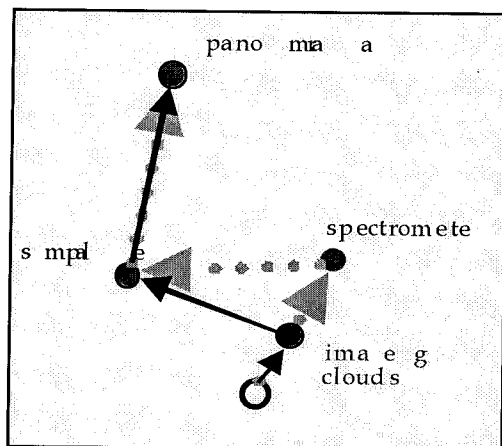


Figure 2: Actual execution example.

5 Implementation Notes

Although the science-based executive was fully functional and all interfaces with the ControlShell finite state machines were fully implemented, some portions of the demonstration were simulated.

The common lisp and ESL executive implementation was not run in the Rocky7 on-board VxWorks environment, but alternatively from a Sparc workstation and from a Macintosh powerbook, communicating over a wireless ethernet. The VxWorks lisp implementation used for the Deep Space One mission's Remote Agent Experiment was too large for memory- and power-constrained Rocky7. A compact implementation of common lisp for VxWorks, capable of supporting ESL, has since been implemented [6].

Generation of power and sun sensor voltage telemetry messages from the ControlShell intelligent behavior software was not implemented; these events were issued via manually invoked NDDS productions. The executive side of this interface was complete, correctly processing the voltage telemetry messages.

The science-based executive and planning assistant implementation is compact, consisting of eleven science request functions, nineteen planning assistant functions, and a thirteen-line main executive function. The bulk of the software implements the command and telemetry interface with the Rocky7 finite state machines.

6 Conclusion

A simple but powerful executive for control of planetary robotic vehicles, capable of accepting science requests directly, has been implemented, tested, and demonstrated with the Rocky7 research vehicle. This was greatly facilitated by ESL's constructs for coordinating multiple software tasks, and by Rocky7's hierarchy of robust behaviors with articulate feedback.

Acknowledgements

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

The author would like to express special thanks to Erann Gat for his help with understanding and using ESL.

References

- [1] R. Volpe, J. Balaram, T. Ohm, R. Ivlev. "The Rocky 7 Mars Rover Prototype". In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, November 4-8 1996, Osaka, Japan.
- [2] *ControlShell User's Manual*, Version 5.1, Real-Time Innovations, Inc., Sunnyvale, California, January 1996.
- [3] E. Gat, The ESL user's guide, Version 0.8, Jet Propulsion Laboratory, internal document, 20 August 1997.
- [4] B. Pell, E. Gat, R. Keesing, N. Muscettola, and B. Smith, "Execution for Autonomous Spacecraft," In *Working Notes of the AAAI Fall Symposium on Plan Execution*, Cambridge, Massachusetts, 1996.
- [5] *Network Data Delivery Service User's Manual*, Version 1, Real-Time Innovations, Inc., Sunnyvale, California, November 1994.
- [6] E. Gat, email communication, 2 February 2001.